CLAIMS

1. A method for high speed interprocess communications comprising the steps of:

attaching first and second processes to a message buffer in a shared region of random access memory (RAM) exclusive of operating system kernel space, each said process having a message list;

accumulating message data from said first process in a location in said message buffer;

adding to said message list of said second process a memory offset corresponding to said location in said message buffer; and,

processing in said second process said accumulated data at said location corresponding to said offset,

whereby said accumulated message data is transferred from said first process to said second process with minimal data transfer overhead.

2. The method according to claim 1, wherein the attaching step comprises the steps of:

detecting a previously created shared region of RAM;

if a shared region of RAM is not detected, creating and configuring a shared region of RAM for storing accumulated data; and,

attaching said first and second processes to said shared region.

3. The method according to claim 1, wherein said message list is a message queue.

4. The method according to claim 3, wherein the adding step comprises the steps of:

retrieving a memory offset in said message buffer corresponding to said

location of data accumulated by said first process; and,

inserting said memory offset in said message queue corresponding to said second process.

5. The method according to claim 4, wherein the inserting step comprises the step of atomically assigning said memory offset to an integer location in said message queue corresponding to said second process.

6. The method according to claim 1, wherein the processing step comprises the steps of:

identifying a memory offset in said message list corresponding to said second process;

processing in said second process message data stored at a location in said message buffer corresponding to said memory offset; and,

releasing said message buffer.

7. A method for configuring high speed interprocess communications between first and second processes comprising the steps of:

disposing a message buffer in a shared region of random access memory (RAM) shared between said first and second processes;

accumulating message data from said first process in a location in said message buffer;

adding to a message list corresponding to said second process a memory offset corresponding to said location in said message buffer; and,

processing in said second process said accumulated message data stored in said message buffer at a location corresponding to said offset,

whereby said accumulated message data is transferred from said first process to said second process with minimal data transfer overhead.

8. The method according to claim 7, wherein the disposing step comprises the steps of:

creating and configuring a message buffer in a shared region of RAM exclusive of operating system kernel space; and,

creating a message list in said shared region for each said process,

whereby said message list can store memory offsets of message data stored in said message buffer.

9. The method according to claim 7, wherein said message list is a message queue.

10. The method according to claim 9, wherein the adding step comprises the steps of:

retrieving a memory offset in said message buffer, said memory offset corresponding to said location of said message data accumulated by said first process; and,

inserting said memory offset in said message queue corresponding to said second process.

11. The method according to claim 10, wherein the inserting step comprises the step of atomically assigning said memory offset to an integer location in said message queue corresponding to said second process.

12. The method according to claim 7, wherein the processing step comprises the steps of:

identifying a memory offset in said message list corresponding to said second process;

processing in said second process said accumulated message data at a

location in said message buffer corresponding to said memory offset; and,

releasing said message buffer.

13. A computer apparatus programmed with a set of instructions stored in a fixed medium for high speed interprocess communications, said programmed computer apparatus comprising:

means for attaching first and second processes to a message buffer in a shared region of random access memory (RAM) exclusive of operating system kernel space, each said process having a message list;

means for accumulating message data from said first process in a location in said message buffer;

means for adding to said message list of said second process a memory offset corresponding to said location in said message buffer; and,

means for processing in said second process said accumulated data at said location corresponding to said offset.

14. The computer apparatus of claim 13, wherein the attaching means comprises:

means for detecting a previously created shared region of RAM; and,

means for creating and configuring a shared region in RAM for storing accumulated data if a previously created shared region of RAM is not detected by said detecting means.

15. The computer apparatus according to claim 13, wherein said message list is a message queue.

16. The computer apparatus according to claim 15, wherein the adding means comprises:

means for retrieving a memory offset in said message buffer corresponding to said location of data accumulated by said first process; and,

means for inserting said memory offset in said message queue corresponding to said second process.

17. The computer apparatus according to claim 16, wherein the inserting means comprises means for atomically assigning said memory offset to an integer location in said message queue corresponding to said second process.

18. The computer apparatus according to claim 13, wherein the processing means comprises:

means for identifying a memory offset in said message list corresponding to said second process;

means for using in said second process message data at a location in said message buffer corresponding to said memory offset; and,

means for releasing said message buffer.

add a'